

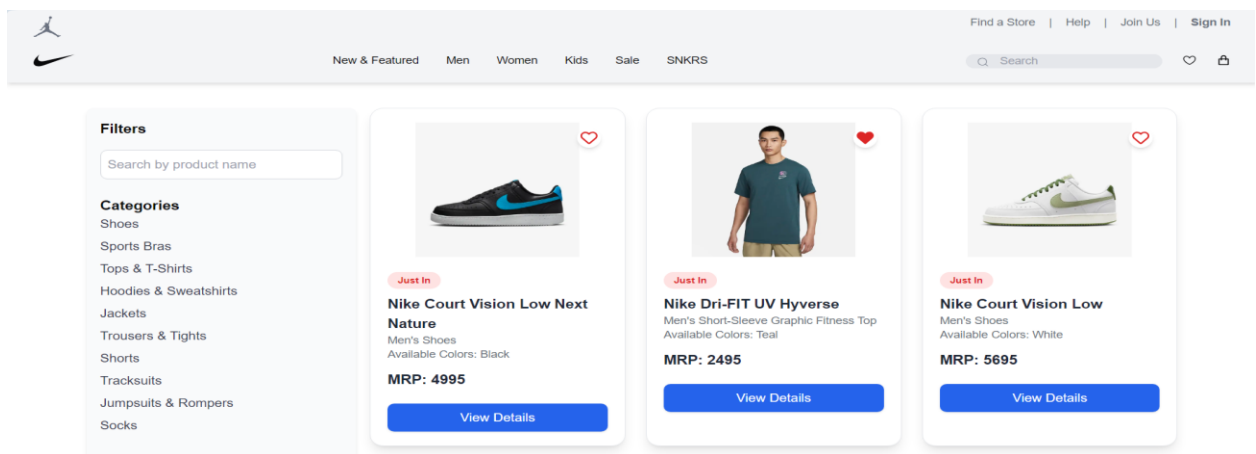
DAY 4: HACKATHON 3

“ Building Dynamic Frontend Components for Your Marketplace”

Key Components:

1. Product Listing:

Snippet:



2. Product Detail (dynamic route):

Code snippet:

```
1 import Product from "@components/Product";
2 import ProductDetail from "@components/ProductDetail";
3 import { client } from "@sanity/lib/client";
4
5 export default async function ProductDetailPage({ params }: { params: Promise<{ id: string }> }) {
6   const productId = (await params).id; // Ensure params.id is extracted correctly
7   console.log("params", productId);
8
9   const query = `
10     *_type == "product" && _id == $productId[0]{
11       _id,
12       productName,
13       "imageUrl": image.asset->url,
14       colors,
15       price,
16       description,
17       status,
18       category,
19       inventory
20     }
21   `;
22   const product = await client.fetch(query, { productId });
23
24   console.log(product)
25
26   if (!product) {
27     return <p>Product not found!</p>;
28   }
29
30   return (
31     <div>
32       <ProductDetail product={product} />
33     </div>
34   );
35 }
```

Snippet:



Nike Dri-FIT UV Hyverse

The Nike Dri-FIT UV Hyverse graphic top offers comfort and UV protection in style. Made from soft, breathable fabric, this fitness top keeps you cool and dry during workouts or casual outings.

Price: Rs.2495

Available Colors:



Add to Cart

3. Searchbar Component:

Code snippet:

```
1  "use client";
2
3  import React from "react";
4  import { useState } from "react";
5
6  const Searchbar: React.FC<{ onSearch: (query: string) => void }> = ({
7    onSearch,
8  }) => {
9    const [searchQuery, setSearchQuery] = useState("");
10
11    const handleSearchChange = (e: React.ChangeEvent<HTMLInputElement>) => {
12      const value = e.target.value; //get the input value
13      setSearchQuery(value); //update the state
14      onSearch(value); //pass value to parent component
15    };
16
17    return (
18      <div className="w-full mb-4">
19        <input
20          type="text"
21          placeholder="Search products"
22          value={searchQuery}
23          onChange={handleSearchChange} //event handler
24          className="w-full p-3 border border-gray-300 rounded-lg text-sm md:text-base"
25        />
26      </div>
27    );
28  };
29
30  export default Searchbar;
```

Snippet:

Filters

Categories

Shoes

Sports Bras

Tops & T-Shirts

Hoodies & Sweatshirts

Jackets



Trousers & Tights

Shorts

Tracksuits

Jumpsuits & Rompers

Socks



Just In

Nike Air Max 270
Men's Shoes
Available Colors: Black

MRP: 13295

[View Details](#)

4. Add to Cart Component:

Code snippet (AddToCart):


```



1  "use client";
2
3  import { useState } from "react";
4  import { client } from "@sanity/lib/client";
5  import React, { useEffect } from "react";
6
7  interface Product {
8    id: string;
9    image: string;
10   name: string;
11   price: number;
12   color: number;
13   type: string;
14   status: string;
15   count?: number;
16 }
17
18 const AddToCart = () => {
19   const [products, setProducts] = useState<Product[]>([]);
20   const [cart, setCart] = useState<Product[]>([]);
21
22   useEffect(() => {
23     const fetchProducts = async () => {
24       const query = `*[ _type == "product" ] {
25         _id,
26         productName,
27         category,
28         price,
29         inventory,
30         colors,
31         status,
32         description,
33         "image": image.asset._ref
34       }`;
35       const fetchedProducts = await client.fetch(query);
36       setProducts(fetchedProducts);
37     };
38     fetchProducts();
39   }, []);
40
41   const addToCart = (product: Product) => {
42     setCart((prevCart) => {
43       const productExists = prevCart.find((item) => item.id === product.id);
44
45       if (productExists) {
46         return prevCart.map((item) =>
47           item.id === product.id
48             ? { ...item, count: (item.count || 1) + 1 }
49             : item
50         );
51       }
52     });
53     return [...prevCart, { ...product, count: 1 }];
54   };
55
56   const removeFromCart = (productId: string) => {
57     setCart((prevCart) =>
58       prevCart
59         .map((item) =>
60           item.id === productId && item.count! > 1
61             ? { ...item, count: item.count! - 1 }
62             : item
63         )
64         .filter((item) => item.count! > 0)
65     );
66   };
67
68   return (
69     <div>
70       <h1>Products</h1>
71       <ul>
72         {products.map((product) => (
73           <li key={product.id} className="flex gap-2 items-center">
74             <div>
75               <h2>{product.name}</h2>
76               <p>${product.price}</p>
77             </div>
78             <button
79               onClick={() => addToCart(product)}
80               className="flex justify-start gap-3 bg-black text-white font-poppins leading-[24px] weight-[500] px-4 py-2 my-3 mx-4 rounded-full hover:bg-gray-300"
81             >
82               Add to Cart
83             </button>
84             <button
85               onClick={() => removeFromCart(product.id)}
86               className="flex justify-start gap-3 bg-black text-white font-poppins leading-[24px] weight-[500] px-4 py-2 my-3 mx-4 rounded-full hover:bg-gray-300"
87             >
88               Remove From Cart
89             </button>
90           </li>
91         ))}
92       </ul>
93
94       </* cart item section */>
95       <h2>Cart Items</h2>
96       {cart.length === 0 ? (
97         <p>No items in the cart.</p>
98       ) : (
99         <ul>
100           {cart.map((item) => (
101             <li key={item.id} className="flex gap-2 items-center">
102               <div>
103                 <h2>{item.name}</h2>
104                 <p>
105                   Rs. {item.price} x {item.count} = Rs.
106                   {(item.price * (item.count || 1)).toFixed(2)}
107                 </p>
108               </div>
109               <button
110                 onClick={() => addToCart(item)}
111                 className="flex justify-start gap-3 bg-green-500 text-white font-poppins leading-[24px] weight-[500] px-4 py-2 my-3 mx-4 rounded-full hover:bg-green-700"
112               >
113                 +
114               </button>
115               <button
116                 onClick={() => removeFromCart(item.id)}
117                 className="flex justify-start gap-3 bg-green-500 text-white font-poppins leading-[24px] weight-[500] px-4 py-2 my-3 mx-4 rounded-full hover:bg-green-700"
118               >
119                 -
120               </button>
121             </li>
122           ))}
123         </ul>
124       )}
125     </div>
126   );
127 }
128
129 export default AddToCart;
130
131
132
133

```

5. Checkout Component:


Snippet:



000 800 100 9538  

How would you like to get your order?

Customs regulation for India require a copy of the recipient's KYC. The address on the KYC needs to match the shipping address. [Learn More](#)

 **Deliver It**


Enter your name and address:

First Name


Last Name

Address Line 1

Order Summary



Nike Air Force 1 React
Qty: 1



Nike Air Force 1 PLT.AF.ORM
Qty: 1

Subtotal:

Rs 21990.00

Final Checklist:

Frontend Component Development:	Styling and Responsiveness:	Code Quality:	Documentation and Submission:	Final Review:
✓	✓	✓	✓	✓